Claims

What is claimed is:

1. A system that facilitates employment of a pluggable formatter, comprising:

a decision module operable to retrieve a first data structure;

a plurality of rule sets that define serialization information about data structure types;

a serialization selector adapted to determine a rule set of the plurality of rule sets to provide to the decision module based on the data structure type, the decision module populating a second data structure based on the serialization information and providing a pluggable formatter with the second data structure, so that the pluggable formatter can serialize the second data structure to an externalized format defined by the pluggable formatter.

2. The system of claim 1, the rule set being definable in the data structure.

3. The system of claim 1, the rule set being defined in a third party file.

4. The system of claim 1, the rule set being a default format based on markings in the first data structure.

5. The system of claim 1, the first data structure being an object.

6. The system of claim 1, the decision module being adapted to retrieve a graph of objects one object at a time for serialization.

7. The system of claim 6, further comprising an object ID generator coupled to the decision module, the object ID generator being adapted to assign object IDs to each object in the graph of objects.

8. The system of claim 1, the data structure containing information within the data structure that the serialization selector utilizes in determining a rule set.

9. The system of claim 1, the decision module being integrated into the pluggable formatter.

10. The system of claim 1, the decision module being separate from the pluggable formatter.

11. A system that facilitates employment of a pluggable formatter, comprising:

a formatter services component adapted to receive a decoded serialized stream from a pluggable formatter and create a data structure for deserialization of the decoded serialized stream; and

an object manager adapted to track data in the decoded serialized stream and determine forward references to additional data to provide fixups to the data structure upon receipt of the additional data.

12. The system of claim 11, further comprising a serialization selector adapted to determine a rule set of a plurality of rule sets to provide to the formatter services component based on the data structure type being deserialized, the formatting services component populating the data structure based on the determined rule set.

13.   The system of claim 12, the rule set being definable in the data structure.

14.   The system of claim 12, the rule set being defined in a third party file.

15.   The system of claim 11, the data structure being an object.

16.   The system of claim 11, the formatter services component being adapted to reinstantiate each object of a graph of objects and the object manager performing fixups on the objects after the graph of objects have been deserialized.

17.   A system for serialization and deserialization of an object that facilitates employment of a pluggable formatter, comprising:

a decision module being operable for retrieving an object from a graph of objects and determining a serialization format for the object based on user defined rules for that object type, the decision module populating a data structure based on the user defined rules and providing the pluggable formatter with the data structure, so that the pluggable formatter can serialize the data structure to a serialized stream in an externalized format defined by the pluggable formatter; and

an object manager adapted to manage the deserialization of objects by the pluggable formatter, the pluggable formatter providing the object manager with a decoded serialized stream.

18.   The system of claim 17, further comprising a serialization selector adapted to select between one of a user defined rule set defined in the object, a user defined rule set defined in another object and a default rule set for determining a serialization format for the object.

19.   The system of claim 18, further comprising a formatter services component adapted to receive a serialized stream of an object and instantiate an object type for deserialization of the decoded serialized stream.

20.   The system of claim 19, further comprising a serialization binder component adapted to determine the object type to instantiate.

21.   The system of claim 20, the object type being one of a similar type of the object being deserialized, a different type and the same type in a different assembly.

22.   The system of claim 17, the object manager being adapted to track forward references to additional objects down the stream and call a callback component so that fixups to the objects can be performed upon receipt of the additional objects.

23.   The system of claim 17, the object manager being adapted to handle remoting procedures.

24.   The system of claim 17, further comprising an object ID generator coupled to the decision module, the object ID generator being adapted to assign a unique object ID to each object in the graph.

25.   The system of claim 17, the system being an architecture that supports both local and remote serialization.

26.   The system of claim 17, further comprising a remoting component for providing the pluggable formatter with remoting information.

27.   The system of claim 26, the remoting component being a user defined rule set located in another object, such that the pluggable formatter

receives one of an object, a remote object and an object reference for serializing in an externalized format, the pluggable formatter being without additional functionality for handling the differences between an object, a remote object and an object reference.

28.   A method for serialization of a graph of objects into a stream, the method comprising:

retrieving an object from the graph of objects;

assigning the object a unique identification number;

selecting a serialization rule set from one of a user defined rule set in the object, a user defined rule set in another object and a default rule set;

pushing the object to a data structure and populating the data structure based on the selected serialization rule set; and

serializing the data structure to an externalized format defined by a pluggable formatter.

29.   The method of claim 28, further comprising retrieving, assigning, selecting, pushing and serializing of each object in a graph of objects.

30.   The method of claim 29, the pushing the object to a data structure and populating the data structure based on the selected serialization rule set being repeated for each object in the graph of objects prior to serializing the data structure to an externalized format defined by a pluggable formatter.

31.   The method of claim 29, the pushing the object to a data structure and populating the data structure based on the selected serialization rule set and serializing the data structure to an externalized format defined by a pluggable formatter being repeated for each object in the graph of objects one at a time.

32.   A method for deserialization of a stream into a graph of objects, the method comprising:

receiving a decoded serialized stream from a pluggable formatter;

retrieving an object from decoded serialized stream;

determining an object type for deserialization;

instantiating an uninitialized instance of the object type;

selecting a serialization rule set from one of a user defined rule set in the object, a user defined rule set in another object and a default rule set; and

populating the uninitialized instance of the object type based on the selected serialization rule set.

33.   The method of claim 32, further comprising registering the object with an object manager.

34.   The method of claim 32, further comprising tracking forward references to other objects within the object.

35.   The method of claim 32, further comprising repeating retrieving, determining, instantiating and populating for each object in an object graph.

36.   The method of claim 35, further comprising performing fixups on the objects in the object graph caused by forward references.

37.   A computer readable medium having computer executable components comprising:

a pluggable formatter component programmed to receive a data structure being populated with serialization information of an object of a graph of objects and serialize the object to an externalized format; and

an object manager component programmed to manage the deserialization of objects by the pluggable formatter component.

39.   The computer readable medium of claim 38, a decision component programmed to retrieve an object and invoke a serialization selection component,

the serialization selection component determining serialization information on the object, the serialization information being user definable in the object through one of methods defined in a serializable class, methods defined in a surrogate class provide by the user and markings provided in the object invoking a default serialization format.

39. The computer readable medium of claim 38, further comprising an object ID generator component programmed to assign a unique object ID to each object in the graph.

40. The computer readable medium of claim 38, the pluggable formatter being programmed to receive a serialized stream and decode the serialized stream.

41. The computer readable medium of claim 40, further comprising a formatter services component adapted to receive the serialized stream of the object and instantiate an object type for deserialization of the serialized stream.

42. The computer readable medium of claim 41, further comprising a serialization binder component adapted to determine the object type to instantiate.

43. The computer readable medium of claim 42, the object type being one of a similar type of the object being deserialized, a different type and the same type in a different assembly.

44. The computer readable medium of claim 43, the object manager component being programmed to track forward references to additional objects of the graph of objects down the stream and call a callback component programmed to perform fixups to the objects upon receipt of the additional objects.

45.   The computer readable medium of claim 38, the pluggable formatter component comprising:

a conversion component programmed to convert the data structure into the external format and decode an object in a serialized stream in the external format;

an object writer programmed to pull object information from the conversion component and push the information to a format writer writing the encoded stream; and

a format parser programmed to read object information from an encoded stream and push the object information to an object reader, the object reader registering the object with the object manager.

46.   The computer readable medium of claim 45, the pluggable formatter further comprising an object information wrapper programmed to handle remoting information.

47.   The computer readable medium of claim 38, the externalized format being XML.

48.   The computer readable medium of claim 38, the externalized format being binary.

49.   A system that facilitates making formatters pluggable, the system comprising:

means for providing serialization information of an object to a pluggable formatter, so that the pluggable formatter can serialize the object in a selectable externalized format;

means for tracking deserialization of the object outside the pluggable formatter; and

means for instantiating an uninitialized instance of an object type, so that the object information can populate the object type.

50.   The system of claim 49, further comprising means for performing fixups on the objects due to forward references.